ELSEVIER

Journal of Statistical Planning and Inference 43 (1995) 381–402

journal of statistical planning and inference

Exploratory designs for computational experiments☆

Max D. Morris*, Toby J. Mitchell

Mathematical Sciences Section, EPM Division, Oak Ridge National Laboratory, Building 6012, P.O. Box 2008, Oak Ridge, TN 37831-6367, USA

Received 22 January 1993

Abstract

Recent work by Johnson et al. (*J. Statist. Plann. Inference* **26** (1990) 131–148) establishes equivalence of the maximin distance design criterion and an entropy criterion motivated by function prediction in a Bayesian setting. The latter criterion has been used by Currin et al. (*J. Amer. Statist. Assoc.* **86** (1991) 953–963) to design experiments for which the motivating application is approximation of a complex deterministic computer model. Because computer experiments often have a large number of controlled variables (inputs), maximin designs of moderate size are often concentrated in the corners of the cuboidal design region, i.e. each input is represented at only two levels. Here we will examine some maximin distance designs constructed within the class of Latin hypercube arrangements. The goal of this is to find designs which offer a compromise between the entropy/maximin criterion, and good projective properties in each dimension (as guaranteed by Latin hypercubes). A simulated annealing search algorithm is presented for constructing these designs, and patterns apparent in the optimal designs are discussed.

Key words: Bayesian prediction; Computer experiment; Computer model; Interpolation; Latin hypercube design: Maximin design; Random functions

1. Introduction

The setting for this paper is that of deterministic function approximation, a subject which has attracted increasing attention in the statistical community in recent years. We shall denote the function of interest by y, and say that it has arguments $x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(k)}$, written collectively as the k-vector $\boldsymbol{x}$. The eventual aim is that of

☆Research sponsered by the Applied Mathematical Sciences Research Program, Office of Energy Research, US Department of Energy under contract DE-AC05-84OR21400 with the Martin Marietta Energy Systems, Inc.

*Corresponding author.

SSDI 0378-3758(94)00035-2

constructing an approximation or prediction of $y(\boldsymbol{x})$ for any $\boldsymbol{x} \in T$, where T is a defined domain or region of interest. In this paper we will limit consideration to $T=[0, 1]^k$, that is, we will suppose that the range of each function argument has been scaled to the unit interval, and that the joint region of interest is the k-dimensional unit cube.

In the kind of application that we consider here, y is often expressed as a computer model, hence the phase *computer experiment*. In this context, y may be thought of as a scalar output (or some chosen scalar function of vector outputs) of the deterministic model which results from specified values of inputs $\boldsymbol{x}$. When y is difficult to evaluate, e.g. requires considerable machine time on an advanced computer system, we may for practical purposes consider y to be an 'unknown' function of the inputs, since classical analysis of the function is often impossibly complicated.

In this paper, we are interested in the design of a computer experiment, comprised of n evaluations of y at selected values of $\boldsymbol{x}$, to serve as the basis for constructing an approximation of y which can be easily evaluated at input values for which the computer model has not been evaluated. Such a 'cheap' surrogate for y is useful in computational activities which require many function evaluations, e.g. maximization or Monte Carlo simulation. Applied research in statistical approaches to the design and analysis of computer experiments for this purpose has been discussed by Sacks et al. (1989) and Currin et al. (1991).

The basis of our work is described in the latter reference, and follows in the spirit of fundamental work in Bayesian function prediction by, for example, Kimeldorf and Wahba (1970) and Micchelli and Wahba (1981). Basically, a 'spatial' stochastic process or random function Y is defined over T as an initial expression of our uncertainty about y. Here, Y will be a stationary Gaussian process for which

$$\operatorname{corr}[Y(\boldsymbol{x}_s),\ Y(\boldsymbol{x}_t)] = R[d(\boldsymbol{x}_s, \boldsymbol{x}_t)],$$

i.e. correlation between responses at two input values (or 'sites', borrowing an intuitive term from geostatistics) is a function of some distance d defined between those two values. In this paper, we shall specifically consider rectangular and Euclidean distance, i.e.

$$d(\boldsymbol{x}_s, \boldsymbol{x}_t) = \sum_{l=1}^{k} |x_s^{(l)} - x_t^{(l)}|, \qquad d(\boldsymbol{x}_s, \boldsymbol{x}_t) = \left[\sum_{l=1}^{k} (x_s^{(l)} - x_t^{(l)})^2 \right]^{1/2},$$

respectively. Once a design of n runs has been specified, and the corresponding evaluations of y made, completion of a prediction of y at any $\boldsymbol{x}$ is straightforward: Bayes' theorem leads to a posterior process, which is also Gaussian, and squared error loss leads to the use of the posterior mean function (of $\boldsymbol{x}$) as the prediction or approximation, $\hat{y}(\boldsymbol{x})$. In addition, the variance or standard deviation of the posterior process at a particular site can be thought of as a measure of 'predictive uncertainty' at that site.

The remainder of this paper is concerned with the construction of experimental designs for function prediction, loosely motivated by the general prediction methodology just described.

2. An optimality criterion

Given a definition of distance, the following notation is useful in discussing our criterion for ranking designs. For a given design D, define a *distance* list $(d_1, d_2, \ldots, d_m)$ in which the elements are the distinct values of inter-site distances, sorted from the smallest to the largest. Hence m can be as large as $\binom{n}{2}$ or as small as 1. Also, define an *index* list $(J_1, J_2, \ldots, J_m)$, in which J_j is the number of pairs of sites in the design separated by distance d_j. Hence the sum of elements of the index list must be $\binom{n}{2}$.

Johnson et al. (1990) explored several connections between certain statistical and geometric properties of designs; in our context, one of their results can be stated as follows. Let $Y_\theta, \theta = 1, 2, 3, \ldots$, be a sequence of stationary Gaussian stochastic processes over T, which differ only in that the correlation function for Y_θ is $R_\theta(\boldsymbol{x}_s - \boldsymbol{x}_t) = [R(d(\boldsymbol{x}_s, \boldsymbol{x}_t))]^\theta$. Here R is a fixed correlation function which must be decreasing in d. The result may then be briefly stated as follows: As θ tends to infinity, the designs that minimize the generalized variance of the posterior process at any finite collection of sites not observed are necessarily those for which (1) d_1 is maximized, and among the designs for which this is true, (2) J_1 is minimized. The authors referred to designs which have this property as maximin (Mm) designs of minimum index. The result then establishes a connection between the geometric Mm criterion, and what might be called a 'D-optimal' prediction criterion in a limit as local correlations become weak.

Here, we shall extend the definition of a maximin design in a somewhat arbitrary but intuitively appealing manner as follows. Call D a maximin design if among available designs, it

(1a) maximizes d_1, and among designs for which this is true;
(1b) minimizes J_1, and among designs for which this is true;
(2a) maximizes d_2, and among designs for which this is true;
(2b) minimizes J_2, and among designs for which this is true;
⋮
(ma) maximizes d_m, and among designs for which this is true;
(mb) minimizes J_m.

Because statements (1a) and (1b) alone specify the definition of Johnson et al. (1990) of a Mm design, our more elaborate definition for Mm optimally essentially only breaks ties among multiple designs which would be Mm (and of minimum index) by their definition. Although this extended definition of Mm is intuitively appealing, we have not established connections between it and asymptotic statistical optimality of higher order.

In the following, it will be necessary to have a scalar-valued design criterion function which can be used to rank competing designs in such a way that the Mm design receives the highest ranking. For this, we introduce a family of functions

$$\phi_p(D)=\left[\sum_{j=1}^{m} J_j d_j^{-p}\right]^{1/p}, \tag{2.1}$$

where p is a positive integer, and J_j and d_j characterize the design D. Note that for large enough p, each term in the sum in (2.1) dominates all subsequent terms, and so from any design class, the designs that minimize ϕ are the Mm designs in that class. The issue of selecting a value of p for a given numerical search will be discussed later.

3. A class of designs

Although the Mm criterion has much intuitive appeal, and the work of Johnson et al. (1990) provides a sound theoretical justification for its use, there are certain characteristics of many unconstrained Mm designs which may not be desirable for computational experiments in practice. Consider, for example, the case in which $n=k+1$, and n is a multiple of 4. For these problems, it can be shown that orthogonal arrays such as Plackett–Burman designs are Mm for $T=[0,1]^k$, with respect either to rectangular or Euclidean distance. However, it is often the case that only one or a few input variables have nonnegligible influence on y, and that the effects of these inputs can be nonlinear. In such circumstances, a two-level design 'collapses' to yield only a few data points, with no response information at intermediate values of any input variable. Of course, the same can be said of many physical experiments; Box and Meyer (1986) use the phrase 'effect sparsity' to describe physical experiments in which a relative few controlled factors are important. However, in physical experiments, y generally includes a random noise component, and a collapsed two-level design provides replication which is useful from the standpoint of estimation. Since (by our definition) computational experiments involve no random noise, even this benefit is lost in this context.

The work reported here is a first step toward developing computational experiment designs which are good both in situations of 'effect sparsity' and in situations where all or most inputs are important. Here, we shall attempt to accomplish such a 'compromise' by applying the Mm criterion within a class of designs for which each one-dimensional projection (i.e. 'collapsed' image) is Mm, as follows. For each of the k design variables, the n scaled values to be used in the experiment are elements of the set $V=\{0, 1/(n-1), 2/(n-1), \ldots, 1\}$. Designs in the class to be investigated assign some ordering of these values to each input variable for the n runs in the experiment, i.e. each column of the n-by-k design matrix contains some permutation of the elements of V. Where these permutations are selected randomly, the result is similar to what Patterson (1954) called a lattice sample, and is a special case of what McKay et al. (1979) called a Latin hypercube sample. Here, we will not discuss randomized

1.) Data.
Input t_0, I_{max}, FAC_t, p.

2.) Initializations.
Randomly select design D from the class.
Set D_{best} to D.
Set t to t_0.

3.) Temperature loop.
Set FLAG to 0.
Set I to 1.

4.) Perturbation loop.
Set D_{try} to D.
Exchange two randomly selected elements in a randomly selected column of D_{try}.

5.) If $\phi_p(D_{try}) < \phi_p(D)$, or with probability $e^{-(\phi_p(D)-\phi_p(D_{try}))/t}$,
Set D to D_{try}.
Set FLAG to 1.

6.) If $\phi_p(D_{try}) < \phi_p(D_{best})$,
Set D_{best} to D_{try},
Set I to 1.
Else,
Increment I by 1.

7.) If $I < I_{max}$,
Branch to step 4.

8.) If FLAG = 1,
Multiply t by FAC_t.
Branch to step 3.

9.) Stop and report D_{best}.

Fig. 1. Search algorithm.

design, but will use this structure as a class from which we hope to identify (fixed) Mm designs; we shall refer to these as maximin Latin hypercube (MmLh) designs. (Park 1991, also considered the class of Latin hypercube arrangements in an optimal design setting using a different selection criterion.)

4. A design construction algorithm

The algorithm described in Fig. 1 is a version of the Metropolis algorithm, or 'simulated annealing'. (A discussion and example of the use of optimization by

simulated annealing in statistical design problems is given by Bohachevsky et al. 1986.) Briefly, a search begins with a randomly chosen Lh design, and proceeds through examination of a sequence of designs, each generated as a *perturbation* of the preceding one. In this case, a perturbation D_{try} of a 'current' design D is formed by interchanging two randomly chosen elements within a randomly chosen column of the corresponding design matrix. (For simplicity, we shall use the notation D to denote both a design and its associated design matrix.) In the course of the search, any time a perturbation of the current design leads to an improvement, i.e. has a lower value of ϕ then the current design, it is adopted as the new current design from which the next perturbation is generated. If a perturbation of the current design leads to a worse design, a random decision is made either to discard the perturbation and retain the current design, or to replace D with D_{try}. In this case, replacement occurs with probability

$$\pi = \exp\{-[\phi(D_{\text{try}}) - \phi(D)]/t\},$$

where t is an algorithm parameter known as the 'temperature', a term which comes from the original physical motivation for annealing as an optimization process. Hence perturbations which lead to slightly worse designs (as measured by ϕ) are more likely to replace the current design than perturbations which lead to significantly worse designs. Also, a given increase in ϕ (decrement in design preference) is more likely to be accepted early in the search when the temperature has a relatively high value, than it is later in the search as the temperature is 'cooled'. This randomized behavior is intended to provide a means by which the search may escape from designs which are only locally optimal, i.e. cannot be improved with respect to ϕ by any single perturbation. Throughout the search, the algorithm keeps track of the 'best' design encountered to date, D_{best}. After a given number of perturbations have been tried at a given temperature without improving the best design, the temperature is lowered by a standard factor and the search continues. When, after a large fixed number of tries, no exchange of the current design for a perturbed design results in an improvement (lower value of ϕ) at a given temperature, the search is ended and D_{best} is reported.

In order to implement the annealing algorithm, the values of certain algorithm parameters must be set. As in most applications of annealing for optimization, we do this by a combination of experience and heuristic rules which seem to work well. Below, we list some brief guidelines which we currently use in setting these values.

Choice of t_0: This is the initial value of t, the temperature parameter. For rectangular distance, t_0 is chosen through a heuristic argument applied to a hypothetical design with a distribution of inter-site distances which is uniform between 50% and 150% of the average inter-site distance. This range can be determined before the search because average inter-site distance for this class of designs is completely determined by the values of n and k. The value of t_0 is then set so that a perturbation of this hypothetical design which decreases one of the currently smallest intersite distances by a small value δ would be accepted with high probability; we currently use

$\delta = 1/(n-1)$, the smallest possible inter-site distance along any one coordinate axis. Specifically, we determine the value of t for which such a perturbation would be accepted with probability $\pi = 0.99$. A similar procedure is used for determining t_0 for a search involving Euclidean distance, except that in this case, we actually perform the search using the square of distance to allow for use of relatively fast integer arithmetic in computing. The hypothetical design has a uniform distribution of squared Euclidean inter-site distances for which the average is again determined by n and k, and we use $\delta = 1/(n-1)^2$ in selecting a value for t_0. As mentioned above, these rules are heuristic but seem to work fairly well in practice; other rules might be more effective. The goal is to begin with a temperature high enough so that the algorithm performs a nearly unbiased random walk among candidate designs early in the search, but not so high that it wastes too much time doing this.

Choice of FAC_t: This is the factor by which the temperature is modified, i.e. a value of 0.95 results in a 5% reduction in temperature. Values slightly less than one yield relatively slow decreases in temperature (relatively slow, relatively successful searches), while smaller values yield more rapid decreases (relatively fast, relatively less successful searches). We have not attempted to develop a rule-of-thumb for determining this value for specific problems, but have used 0.90 and 0.95 with generally good results.

Choice of $I_{\max}$: This parameter is the number of design perturbations the algorithm will try before going on to the next temperature, provided no new best design is found. Whenever a new best design is identified, the counter is reset so that $I_{\max}$ additional perturbations are tried at that temperature. It seems reasonable that $I_{\max}$ should be larger for larger problems, i.e. those for which the number of possible perturbations, $\binom{n}{2} \times k$, is relatively large. We have used about 10 times this number for the value of $I_{\max}$ in many optimizations.

5. Choice of a specific criterion function (p)

The algorithm as outlined above is used to search for a design which is ϕ_p optimal, for a specified value of p. Recall that the argument for use of ϕ_p as a criterion function is based on the fact that, *for large enough* p, it ranks designs for a given problem in the same way that the more cumbersome maximin criterion (statements (1a)–(mb)) does. A practical issue is then how large p must be. Figure 2 displays the ranking of designs for the problem in which $n = 5$, $k = 3$, and rectangular distance is used, based both upon ϕ_p for three values of p, and ϕ_∞, i.e. the true maximin ranking. (A small problem is used here so that all 142 unique designs could be represented on the graphs.) In this case, when $p = 1$ the best design by the ϕ criterion does not coincide with the Mm design. For $p = 2$, although there is some difference between how designs are ranked, the two criteria rank the same design as best. Finally, it seems clear that p as small as 5 is sufficient for purposes of searching for an optimal design in this problem. This

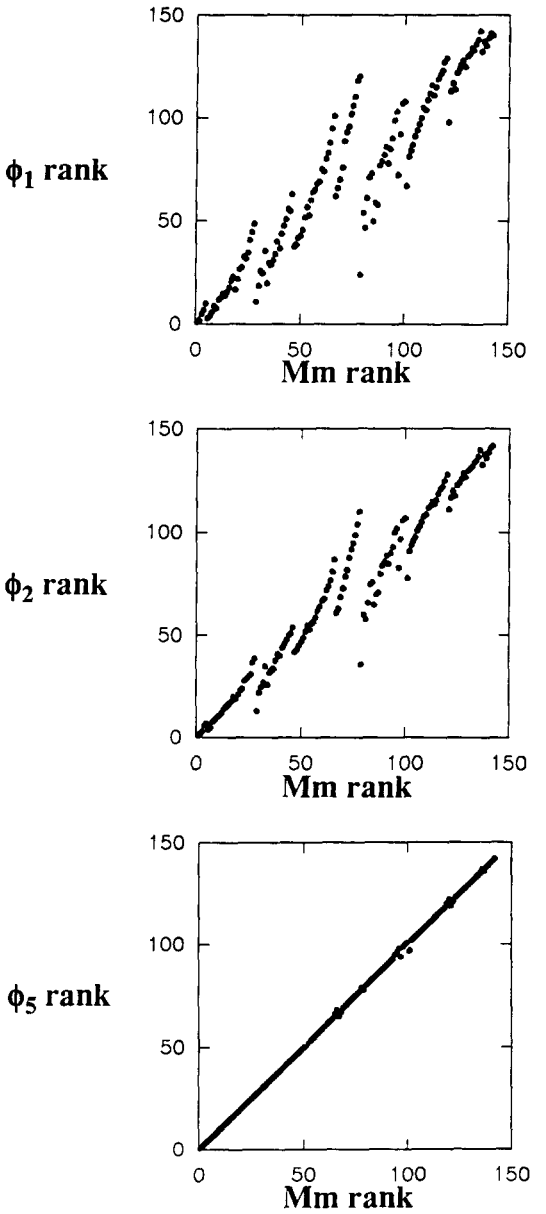

Fig. 2. Comparison of design rankings for ϕ_p and Mm; rectangular distance, $n=5$, $k=3$.

varies greatly with the specific problem, however. Often, but not always, larger problems (those defined by larger values of n and k) require values of p as large as 20–50 before the best design found ranks best among with respect to both the Mm and ϕ criteria.

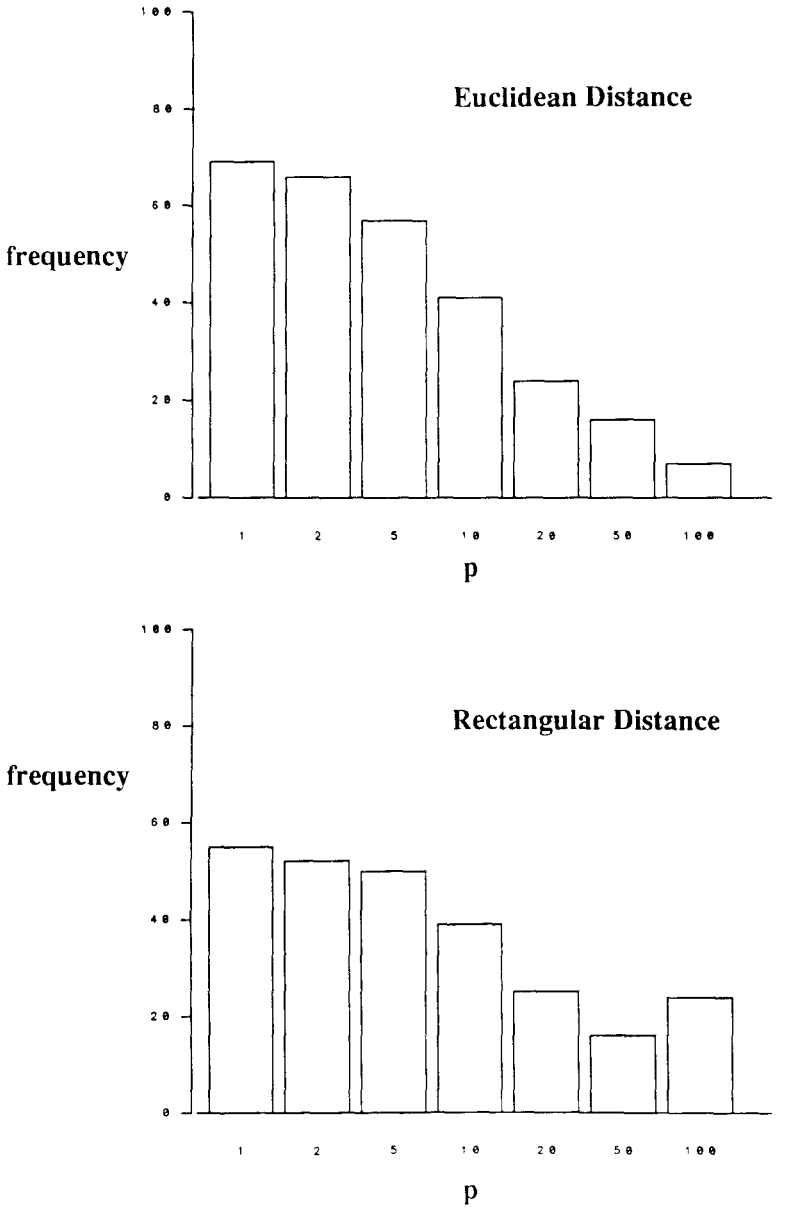

Fig. 3. Number of times Mm design is found in 100 attempts using ϕ_p; $n=8$, $k=4$.

There is, however, another consideration regarding choice of p. The algorithm described above tends to more reliably find a design which minimizes ϕ_p when p is set to a relatively small value. Figure 3 contains a histogram showing the fraction of searches (out of 100 tries) which found the ϕ_p-optimal design when $n=8$ and $k=4$, for

several values of p. Clearly, smaller values of p lead to greater success rates in this problem (and others). In this case, designs which minimized ϕ_p for each value of p examined also minimized ϕ_∞, i.e. even the ϕ_1-optimal design is a Mm design.

These two observations imply that the best value of p would be the smallest one for which ϕ_p and the Mm criterion agree at least on which design is best. In practice, this seems impossible to predict in advance. One can envision relatively straightforward heuristic sequential strategies which would attempt to identify a good value of p in a sequence of several optimizations; to date we have not tried this. Our current approach is to simply use the annealing algorithm to perform several searches at each of several values of p; we currently use $p=1, 2, 5, 10, 20, 50$, and 100. The optimized design which is best with respect to the maximin criterion is selected as the single product of the exercise.

6. Some results

The algorithm described above has been used to generate a catalog of MmLh designs (Morris and Mitchell, 1992) for Euclidean and rectangular distance for each combination of n between 3 and 12 and k between 2 and 5, as well as additional designs for $k=2$ (n–20), $k=n$ (n–9), and $k=n/2$ (n–14). The algorithm described above was used in generating most of these designs. However, in some cases for which it was computationally practical, the design listed was found by complete search over all Latin hypercube arrangements; this was done for $k=2$ with $n\leqslant 11$, $k=3$ with $n\leqslant 6$, and $k=4$ with $n\leqslant 5$.

Tables 1(A) and 1(B) show the smallest value of p for which the apparently optimal design was found, and the fraction of tries at that value of p which resulted in that or an equivalent design. As can be seen from these tables, there are many cases in which the best design found was produced in only a small number of optimizations. In some of these cases, it may very well be that better designs exist.

Tables 2(A) and 2(B) contain values of the minimum distance between pairs of sites in each design, and the index J_1, i.e. the number of pairs of separation d_1. It is useful to remember that the largest possible distance between two sites is k for rectangular distance and $k^{1/2}$ for Euclidean distance.

Selected designs with $k=2$ are graphed in Figs. 4 and 5 for Euclidean and rectangular distances, respectively. The sites for these designs, particularly for larger values of n, are generally equally spaced along parallel lines; the Latin hypercube structure implies that these lines cannot be parallel with either axis. The designs displayed for $n=17$ are identical for the two distance measures, and similar for $n=9$, while the definition of distance seems to be more important for the designs of $n=5$ and 13 points. Figures 6 and 7 display the two-dimensional projections of two designs of higher dimension. The design of Fig. 6 is for $n=12$ and $k=6$, generated using Euclidean distance, and Fig. 7 displays a design for $n=10$ and $k=4$, generated using

Table 1(A)
Smallest value of p for which the apparently optimal design was found, and the fraction of tries at that value of p which resulted in that or an equivalent design; Euclidean distance

n	k							
	2	3	4	5	6	7	8	9
3	a	a	a	1				
	a	a	a	50/50				
4	a	a	a	1				
	a	a	a	25/25				
5	a	a	a	1				
	a	a	a	21/25				
6	a	a	2	1	2			
	a	a	1/10	44/50	2/25			
7	a	5	20	1		2		
	a	10/10	1/10	3/50		1/100		
8	a	50	1	20			10	
	a	1/100	5/10	2/50			1/25	
9	a	20	100	20				1
	a	6/25	2/25	1/25				1/25
10	a	5	5	1				
	a	1/25	1/50	1/15				
11	a	50	50	100				
	a	8/100	1/100	1/100				
12	1	1	50	20	5			
	8/10	55/100	1/100	1/100	1/25			
13	5							
	1/10							
14	1					20		
	7/10					1/100		
15	5							
	4/10							
16	5							
	1/10							
17	1							
	8/10							
18	5							
	1/10							
19	20							
	2/10							
20	100							
	1/10							

[a] Designs found by complete search.

Table 1(B)
Smallest value of p for which the apparently optimal design was found, and the fraction of tries at that value of p which resulted in that or an equivalent design; rectangular distance

n	k							
	2	3	4	5	6	7	8	9
3	a	a	a	1				
	a	a	a	50/50				
4	a	a	a	1				
	a	a	a	25/25				
5	a	a	a	1				
	a	a	a	50/50				
6	a	a	1	1	1			
	a	a	10/10	47/50	6/25			
7	a	1	1	1		1		
	a	10/10	6/10	37/50		1/25		
8	a	1	1	10			5	
	a	10/10	5/10	1/50			2/25	
9	a	10	20	5				1
	a	18/25	1/25	1/25				1/25
10	a	20	10	5				
	a	3/25	3/50	1/15				
11	a	20	20	100				
	a	5/100	1/100	1/100				
12	1	10	10	1	5			
	10/10	4/100	2/100	11/100	6/500			
13	5							
	5/10							
14	2					20		
	9/10					1/100		
15	1							
	10/10							
16	10							
	5/10							
17	2							
	9/10							
18	20							
	5/10							
19	10							
	1/10							
20	1							
	5/10							

[a] Designs found by complete search.

Table 2(A)
Smallest intersite distance and number of pairs separated by that distance; Euclidean distance

n	k							
	2	3	4	5	6	7	8	9
3	0.7071 1	1.2247 3	1.3329 2	1.4142 1				
4	0.7454 4	0.8165 1	1.1547 2	1.2472 1				
5	0.5590 4	0.8292 4	0.9682 1	1.2247 5				
6	0.4472 3	0.7483 4	0.9381 2	1.1314 4	1.2649 6			
7	0.4714 4	0.6872 3	0.8819 4	1.0541 10		1.3017 2		
8	0.4041 4	0.6547 12	0.9258 24	1.0102 4			1.3628 1	
9	0.3953 12	0.5863 4	0.8101 6	0.9763 5				1.4031 1
10	0.3514 7	0.5774 3	0.7857 12	1.0062 20				
11	0.3162 6	0.5385 1	0.7416 2	0.8944 2				
12	0.3278 16	0.5455 6	0.7216 2	0.8672 1	1.0679 4			
13	0.3005 17							
14	0.3172 20					1.1384 4		
15	0.2945 18							
16	0.2749 14							
17	0.2652 12							
18	0.2496 12							
19	0.2357 9							
20	0.2233 5							

rectangular distance. The two-dimensional projections of both designs are fairly well distributed over the unit square, and have interesting symmetry properties.

An interesting property shared by some of the designs is the tendency of design sites to be approximately or exactly equidistant from the center of T. For the designs

Table 2(B)
Smallest intersite distance and number of pairs separated by that distance; rectangular distance

n	k							
	2	3	4	5	6	7	8	9
3	1.0000 1	2.0000 3	2.5000 2	3.0000 1				
4	1.0000 4	1.3333 1	2.0000 2	2.6667 5				
5	0.7500 4	1.2500 3	1.7500 2	2.5000 10				
6	0.6000 3	1.2000 6	1.6000 2	2.2000 5	2.8000 15			
7	0.6667 12	1.0000 2	1.6667 16	2.0000 3		3.0000 8		
8	0.5714 12	1.0000 6	1.5714 16	2.0000 17			3.2857 6	
9	0.5000 8	1.0000 19	1.3750 3	1.8750 11				2.4615 9
10	0.4444 6	0.8889 8	1.3333 5	1.8889 20				
11	0.4000 3	0.8000 2	1.3000 8	1.8000 31				
12	0.4545 16	0.8182 9	1.2727 12	1.7273 10	2.1818 12			
13	0.4167 16							
14	0.3846 13					2.4615 4		
15	0.3571 10							
16	0.3333 5							
17	0.3750 36							
18	0.3529 37							
19	0.3333 29							
20	0.3158 15							

computed here, this property holds primarily among those designs for which $n=k$ or $n=2k$. Tables 3(A) and 3(B) give the minimum and maximum distances between a design site and the center of T for each design.

The designs for which $n=2k$ have other interesting properties. In particular, all such designs tabulated, except in the case of $n=14$ based on rectangular distance, are

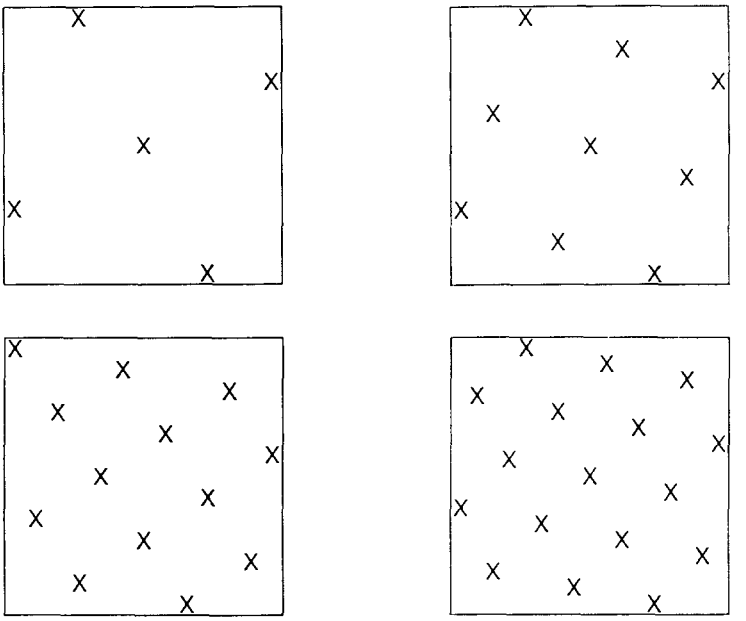

Fig. 4. MmLh designs for $k=2$ and $n=5, 9, 13$, and 17; Euclidean distance.

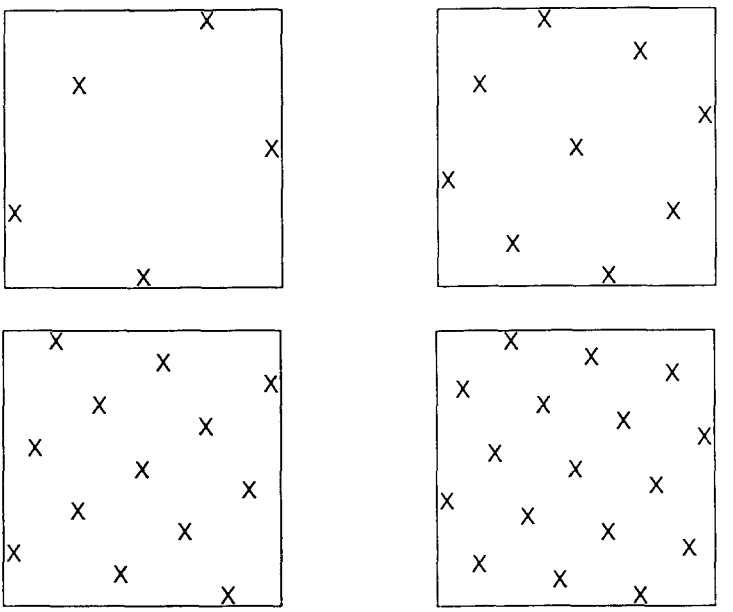

Fig. 5. MmLh designs for $k=2$ and $n=5, 9, 13$, and 17; rectangular distance.

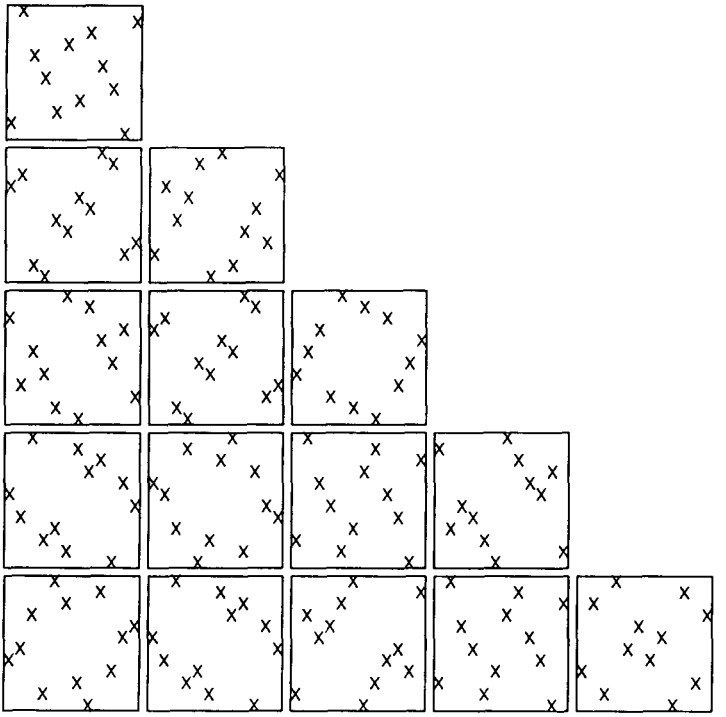

Fig. 6. Two-dimensional projections of MmLh design for $k=6$ and $n=12$; Euclidean distance.

foldover designs. By this, we mean that the design can be partitioned into pairs of sites, each with the property that one site is the reflection through the center of T of the other site. Another interesting geometric property holds for several of these designs. If one site from each foldover pair is eliminated, leaving a k-by-k design matrix, and each element of this reduced matrix is replaced by the absolute difference between the element and $\frac{1}{2}$ (i.e. distance to the center of the region in one-dimensional projections), the elements of the resulting matrix each take one of k unique values. Each of these values appears once in each row and once in each column of the matrix; that is, the design matrix reduced in this way contains k 'symbols' in the pattern of a Latin square.

A similar pattern holds for the designs generated with $n=k=3$–6, for either definition of distance. In these cases, if each element in the entire n-by-n design matrix is replaced by the absolute difference between the element and $\frac{1}{2}$, the resulting matrix contains each unique value twice in each row and column for even n. For odd n, the result is the same except that the zeroes (corresponding to elements which were $\frac{1}{2}$ in the original design matrix) appear only once in each row and column.

We have not undertaken a thorough investigation of the geometric properties of these designs. However, we find the observations noted above to be interesting, and believe that further study might reveal clues as to how 'recipes' for maximin or

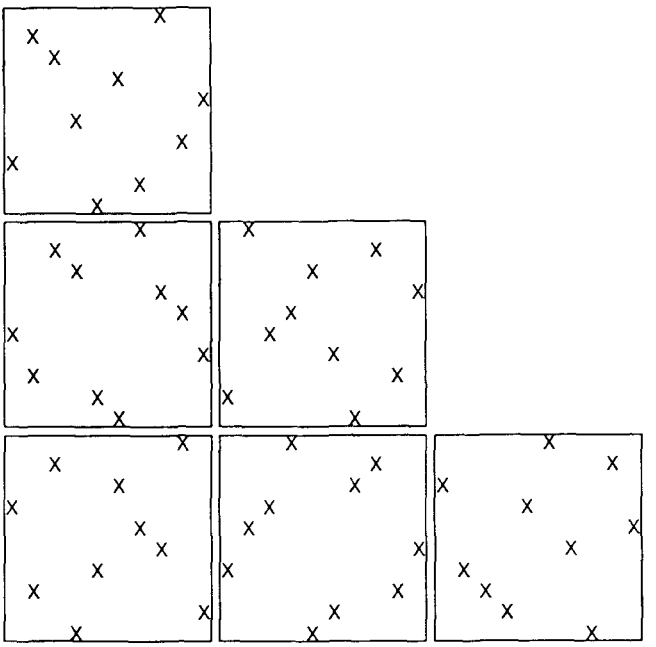

Fig. 7. Two-dimensional projections of MmLh design for $k=4$ and $n=10$; rectangular distance.

near-maximin Latin hypercube designs might be easily constructed without extensive numerical searches.

7. A demonstration

As described earlier, motivation for the designs reported here comes from (1) the asymptotic connection between the maximin the entropy criteria, as established by Johnson et al. and (2) the need to insure that the design will be effective even if the response variable is sensitive to only a few design variables. The following small example demonstrates that, at least in one case, a MmLh design yields results which are superior to those obtained by either a maximin distance design or a randomly chosen Latin hypercube.

The test function used in this exercise is defined by the equation:

$$y=\frac{2\pi z^{(3)}(z^{(4)}-z^{(6)})}{\ln(z^{(2)}/z^{(1)})\left[1+\dfrac{2z^{(7)}z^{(3)}}{\ln(z^{(2)}/z^{(1)})(z^{(1)})^2 z^{(8)}}+\dfrac{z^{(3)}}{z^{(5)}}\right]}. \tag{7.1}$$

The $z^{(i)}$, $i=1,2,\ldots,7$, are unscaled inputs for which meaningful intervals are listed in Table 4; for our purposes the corresponding standardized inputs $x^{(i)}$, $i=1,2,\ldots,7$, are

Table 3(A)
Smallest and largest distances from the center of T to a design site; Euclidean distance

n	k							
	2	3	4	5	6	7	8	9
3	0.5000	0.7071	0.7071	0.8660				
	0.7071	0.7071	0.8660	1.0000				
4	0.5270	0.5528	0.7454	0.7638				
	0.5270	0.7265	0.7454	0.8975				
5	0.0000	0.5590	0.6614	0.7906				
	0.5590	0.7071	0.7500	0.7906				
6	0.1414	0.5196	0.6000	0.7280	0.8367			
	0.5831	0.6557	0.7211	0.7810	0.8367			
7	0.0000	0.4082	0.6009	0.7454		0.8660		
	0.5270	0.7071	0.7265	0.7454		0.8975		
8	0.1010	0.5487	0.6547	0.6888			0.9035	
	0.5440	0.6186	0.6547	0.7457			0.9476	
9	0.0000	0.4677	0.5303	0.6960				0.9520
	0.5590	0.7071	0.7071	0.7603				0.9843
10	0.1757	0.4811	0.5556	0.7136				
	0.5720	0.5958	0.7115	0.7136				
11	0.1414	0.0000	0.5385	0.6403				
	0.5831	0.6481	0.7141	0.7483				
12	0.2318	0.5359	0.5677	0.6508	0.6787			
	0.5183	0.5511	0.6863	0.7565	0.7687			
13	0.1667							
	0.7071							
14	0.1632					0.8059		
	0.5679					0.8276		
15	0.1010							
	0.5759							
16	0.0471							
	0.5676							
17	0.0000							
	0.5590							
18	0.1500							
	0.6294							
19	0.1111							
	0.6334							
20	0.1342							
	0.5966							

scaled and shifted as indicated earlier. The variable $z^{(8)}$ was fixed at a value of 9855 throughout, corresponding to the lower limit of its meaningful range. Although the equation is simple, it has a physical interpretation; y represents steady-state flow of water through a borehole between two aquifers. A more complete

Table 3(B)
Smallest and largest distances from the center of T to a design site; rectangular distance

n	k							
	2	3	4	5	6	7	8	9
3	0.5000	1.0000	1.0000	1.5000				
	1.0000	1.0000	1.5000	2.0000				
4	0.6667	0.8333	1.3333	1.5000				
	0.6667	1.1667	1.3333	1.8333				
5	0.5000	0.7500	1.0000	1.5000				
	0.7500	1.2500	1.2500	1.5000				
6	0.2000	0.9000	1.2000	1.3000	1.8000			
	0.8000	0.9000	1.2000	1.7000	1.8000			
7	0.0000	0.6667	1.0000	1.1667		1.5000		
	0.6667	1.1667	1.3333	1.6667		2.1667		
8	0.1429	0.7857	1.1429	1.2143			2.0000	
	0.7143	1.0714	1.1429	1.5000			2.5714	
9	0.0000	0.7500	0.8750	1.2500				2.1250
	0.6250	1.0000	1.3750	1.5000				2.6250
10	0.2222	0.2778	1.0000	1.3889				
	0.7778	1.1667	1.3333	1.3889				
11	0.2000	0.3000	1.0000	1.2000				
	0.8000	1.1000	1.2000	1.6000				
12	0.2727	0.1364	0.8182	1.3182	1.5455			
	0.7273	1.0455	1.3636	1.5909	1.7273			
13	0.0000							
	0.8333							
14	0.2308					1.6538		
	0.7692					2.1154		
15	0.2143							
	0.7143							
16	0.0667							
	0.8000							
17	0.0000							
	0.7500							
18	0.1176							
	0.9412							
19	0.1667							
	0.8333							
20	0.1579							
	0.7895							

description of the model, including physical units for each variable, is given in Morris et al. (1993).

Table 4 also contains a measure of sensitivity of y to each $x^{(i)}$. With respect to independent, uniform probability measure for each x over the unit interval, the

Table 4
Range and output sensitivity associated with inputs of Eq. (7.1)

Unscaled input	Lower limit	Upper limit	Sensitivity limit
$z^{(1)}$	0.05	0.15	38.91
$z^{(2)}$	100.0	50000.0	0.0596
$z^{(3)}$	63070.0	115600.0	0.000122
$z^{(4)}$	990.0	1110.0	9.550
$z^{(5)}$	63.1	116.0	0.1218
$z^{(6)}$	700.0	820.0	9.550
$z^{(7)}$	1120.0	1680.0	9.527

[a] As defined in Eq. (7.2).

measure is the square root of expected variance associated with a particular x:

$$\sqrt{\underset{x^{(j)},\, j\neq i}{E}\ \underset{x^{(i)}|x^{(j)},\, j\neq i}{\mathrm{Var}}\ y}\,. \tag{7.2}$$

'Effect sparsity' is clearly present, at least to some degree, since $x^{(2)}, x^{(3)}$, and $x^{(5)}$ can account for only very minor variation in y. Inputs $x^{(4)}, x^{(6)}$, and $x^{(7)}$, while somewhat more influential, are considerably less important than $x^{(1)}$. (Note that since the assigned ranges of $z^{(4)}$ and $z^{(6)}$ have the same width, the algebraic form of Eq. (7.1) implies that $x^{(4)}$ and $x^{(6)}$ must be equally influential by this definition.)

Twelve simulated experiments of eight runs were performed to approximate this model. The first design used was a maximin distance design, in this case a Plackett–Burman design for which each $x^{(i)}$, $i=1,2,\ldots,7$, was either 0 or 1 in each run. The second design was a MmLh, computed by the algorithm described for Euclidean distance, and listed in Table 5. The last ten experimental designs used were randomly chosen Latin hypercubes; the only restriction imposed was that each of the values $0, \frac{1}{7}, \frac{2}{7}, \ldots, 1$ was assigned to each $x^{(i)}$ in exactly one of the eight runs. In each simulated experiment, a response surface was fitted to the eight data points using a stationary Gaussian process model, as described by Sacks et al. (1989) and Currin et al. (1991). The correlation function used here was of the form:

$$\mathrm{corr}[Y(\boldsymbol{x}_s), Y(\boldsymbol{x}_t)] = \rho^{\sum_{i=1}^{7} (x_s^{(i)} - x_t^{(i)})^2}.$$

Due to the small sample size, no attempt was made to estimate ρ from the data, but analyses were repeated for ρ of 0.50, 0.90, and 0.98, and given each value of ρ, the corresponding maximum likelihood estimate of $\mu = E(Y)$, denoted by $\hat{\mu}$, was calculated. The fitted response surface was then defined in each case to be:

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}_\rho^{\mathrm{T}} \boldsymbol{R}_\rho^{-1} (\boldsymbol{y} - \hat{\mu}\mathbf{1}),$$

Table 5
MmLh design used in the demonstration (divide each entry by 7)

$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(6)}$	$x^{(7)}$
1	7	4	4	3	6	0
5	0	5	2	1	7	3
0	1	3	3	2	0	2
6	6	7	1	4	1	4
3	2	6	7	7	4	5
4	5	2	6	0	3	7
2	4	1	0	6	5	6
7	3	0	5	5	2	1

Table 6
Root-mean-square error and maximum absolute error for each Mm, MmLh, and random Lh designs, over 1000 random references sites

ρ	Mm	MmLh	average Lh
0.50	20.17	11.40	20.59
	74.57	74.53	100.53
0.90	16.03	11.65	17.38
	47.14	40.14	78.54
0.98	16.18	12.58	17.13
	48.46	44.28	77.00

where $\boldsymbol{r}_\rho$ is the 8-element vector of correlations between Y at $\boldsymbol{x}$ and at each of the design sites, $\boldsymbol{R}_\rho$ is the 8-by-8 correlation matrix among design sites, $\boldsymbol{y}$ is the observed 8-element vector of response values, and $\mathbf{1}$ is an 8-element vector of 1's.

In order to assess the quality of fit of each response surface, 1000 reference values of $\boldsymbol{x}$ were randomly (uniformly) selected from $[0,1]^7$, and values of y calculated and compared to corresponding predictions $\hat{y}$ for each of them. Table 6 displays both the root-mean-squared error and largest absolute error for the maximin distance design, the MmLh design, and averages of these for the 10 random Latin hypercube designs. For each of the three values of ρ used, results are qualitatively similar. The maximum errors of prediction are similar for Mm and MmLh designs, but mean squared error is smaller for the MmLh. On average, the random Latin hypercube designs produced both larger maximum and mean squared errors than did the MmLh; mean squared errors are approximately the same for random Latin hypercubes and the Mm design. Hence, for this problem, the overall performance of the MmLh could be said to be best, while by these measures the performance of the random Latin hypercubes — on average — appears to be worst.

8. Summary

We have suggested the use of maximin Latin hypercube designs for computational experiments in which the general goal is the construction of an approximation or prediction of a deterministic scalar valued output variable as a function of the input variables. Use of the maximin distance criterion is motivated by a result of Johnson et al. (1990), who established an equivalence between the Mm property and a kind of D-optimality for Bayesian prediction, in a limit as local correlations are weakened. Since all Latin hypercube designs (unlike many unconstrained Mm distance designs) are evenly distributed in each one-dimensional projection, they are intuitively

appealing for situations in which only one or few inputs have an important impact on the output variable. Since many computer models display this sort of effect sparsity, the 'compromise' achieved by using the Mm criterion within the class of Latin hypercube arrangements may yield designs which are effective for predicting the output both when few or many inputs are important.

A criterion function, ϕ_p, and a numerical optimization procedure based on simulated annealing have been presented for generating MmLh designs. A catalog of designs produced by this algorithm was given by Morris and Mitchell (1992), and some example designs from that listing were given here. Observations were made concerning some of the geometric properties of the MmLh designs generated, particularly in the cases of $k=2$, $n=k$, and $n=2k$.

References

Bohachevsky, I.O., M.E. Johnson and M.L. Stein (1986). Generalized simulated annealing for function optimization. *Technometrics* **28**, 209–217.

Box, G.E.P. and R.D. Meyer (1986). An analysis for unreplicated fractional factorials. *Technometrics* **28**, 11–18.

Currin, C., T. Mitchell, M. Morris and D. Ylvisaker (1991). Bayesian prediction deterministic functions, with applications to the design and analysis of computer experiments. *J Amer. Statist. Assoc.* **86**, 953–963.

Johnson, M., L. Moore and D. Ylvisaker (1990). Minimax and maxmin distance designs. *J. Statist. Plann. Inference* **26**, 131–148.

Kimeldorf, G.S. and G. Wahba (1970). Spline functions and stochastic processes. *Sankhya Ser. A* **32**, 173–180.

McKay, M.D., W.J. Conover and R.J. Beckman (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245.

Micchelli, C.A. and G. Wahba (1981). Design problems for optimal surface interpolation. In: Z. Ziegler, Ed., *Approximation Theory and Applications*, Proc. Workshop, Technion, Israel Inst. Tech., Haifa, 1980, Academic Press, New York.

Morris, M.D. and T.J. Mitchell (1992). Exploratory designs for computational experiments. ORNL/TM-12045, available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

Morris, M.D., T.J. Mitchell and D. Ylvisaker (1993). Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics* **35**, 243–255.

Park, J.-S. (1991). Tuning complex computer codes to data and optimal designs. Ph.D. Thesis, University of Illinois, Department of Statistics (unpublished).

Patterson, H.D. (1954). The errors of lattice sampling. *J. Roy. Statist. Soc. Ser. B* **16**, 140–149.

Sacks, J., W.J. Welch, T.J. Mitchell and H.P. Wynn (1989). Design and analysis of computer experiments. *Statist. Sci.* **4**, 409–423.